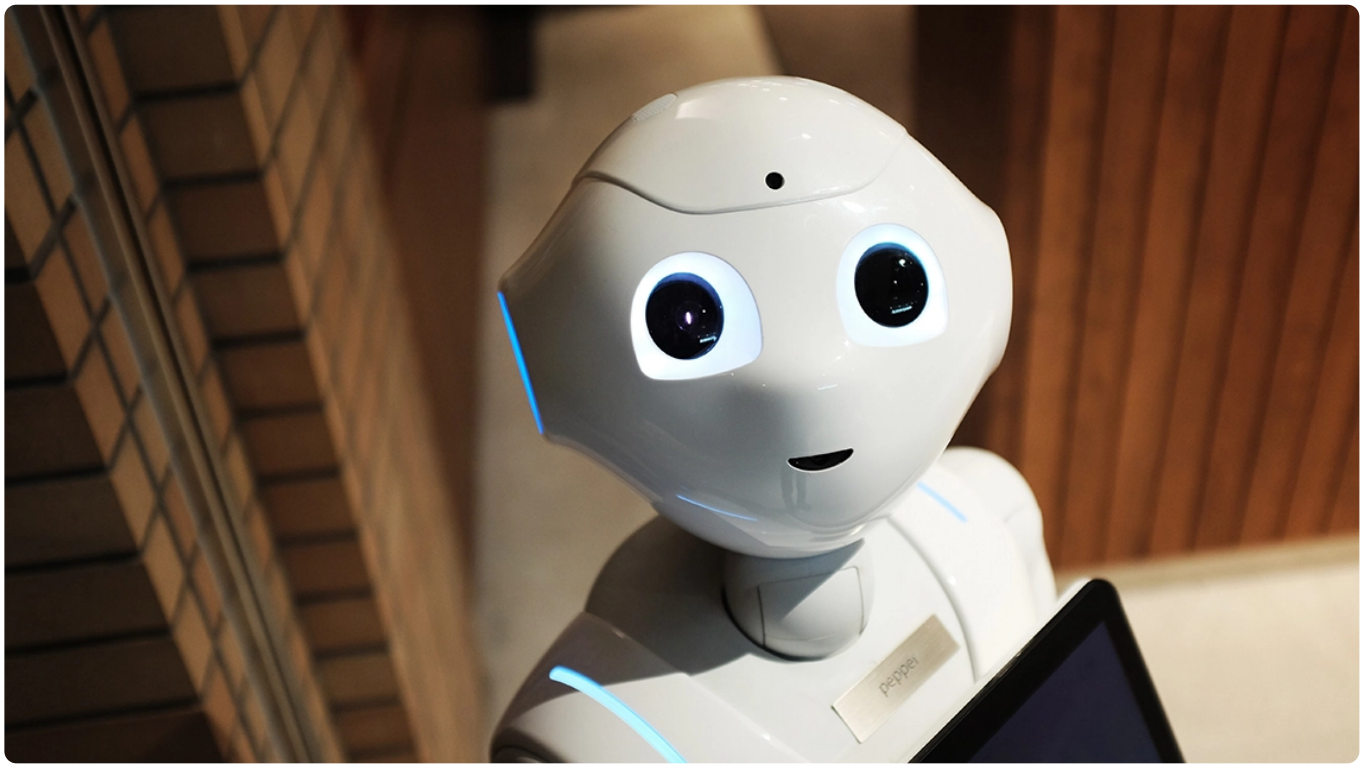
15｜AI变声：音频AI技术的集大成者

2021-12-27 冯建元

《搞定音频技术》 课程介绍 >



讲述：冯建元
时长 11:06 大小 10.18M

你好，我是建元。

AI 技术在音频领域发展十分迅速。除了我们之前讲的降噪、回声消除以及丢包补偿等方向可以用 AI 模型来提升音质听感之外，AI 模型还有很多有趣的应用。其中比较常见的有 ASR（Automatic Speech Recognition）可以理解为语音转文字，TTS（Text To Speech）文字转语音和 VPR（Voice Print Recognition）声纹识别等。

在之前讲音效算法的时候，我们知道，要做到变声需要改变整个语音信号的基频，还需要改变语音的音色。传统算法是通过目标语音和原始语音，计算出基频差距和频谱能量分布的差异等特征，然后使用变调、EQ 等方法来对语音进行调整，从而实现变声（Voice Conversition，VC）。

但这些特征的差异，在发不同的音，不同的语境中可能都是不一样的。如果用一个平均值来进行整体语音的调整，你可能会发现有的音变声效果比较贴近目标语音，而有的音，可能会有比较大的偏离。**整体听感上就会觉得变声效果时好时坏。**

甚至由于某些发音在改变了频谱能量分布后，共振峰发生了较大改变，连原本想表达的语意都发生了变化。所以为了获得比较好的变声效果，我们需要实时对语音做动态的调整，而这使用传统算法显然是无法穷尽所有发音、语境的对应变化关系的。

你可能已经发现了，如图 1 所示，如果我们可以做到语音转文字，文字也可以转语音，那么**结合声纹识别（VPR）把人声的特性加入到 TTS 之中，是不是就可以实现变声的功能了？**

先别急，接下来我们就基于 AI 模型的变声算法的角度，来整体认识一下这些常见 AI 算法背后的原理，以及它们是如何组合、搭配实现变声功能的。

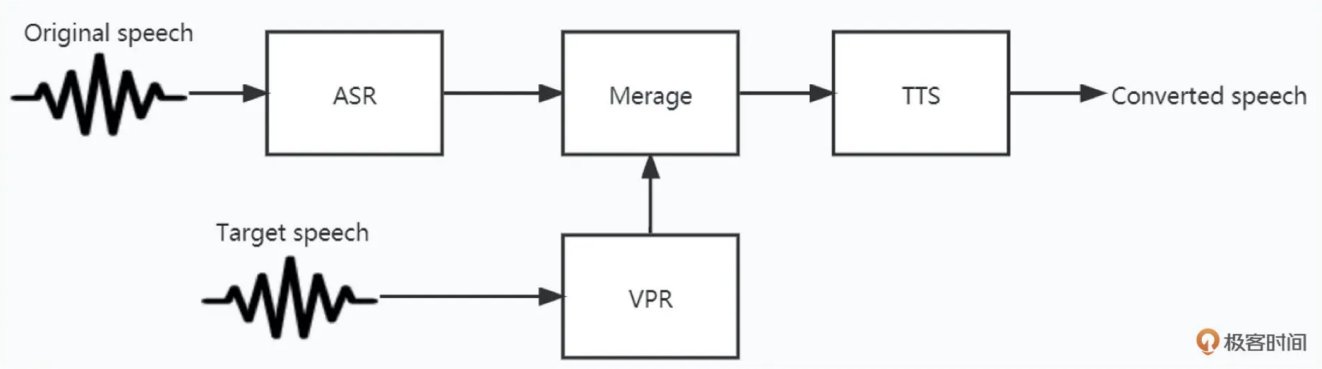


图1 基于ASR、TTS、VPR的变声流程示意图

我们不妨先从 ASR 算法来看看我们是如何实现语音转文字的。

ASR

结合图 2，我们可以看一下 ASR 算法的基本原理。

语音信号需要转换为频谱信号或者我们之前讲过的 MFCC 来作为语音特征序列。然后我们根据特征序列推断出对应的音素序列，音素在不同的语言中有很多不同的表达形式，比如中文可以用汉语拼音来表示。最后根据音素和文字的映射字典（lexicon）就可以得到语言对应的文本了。

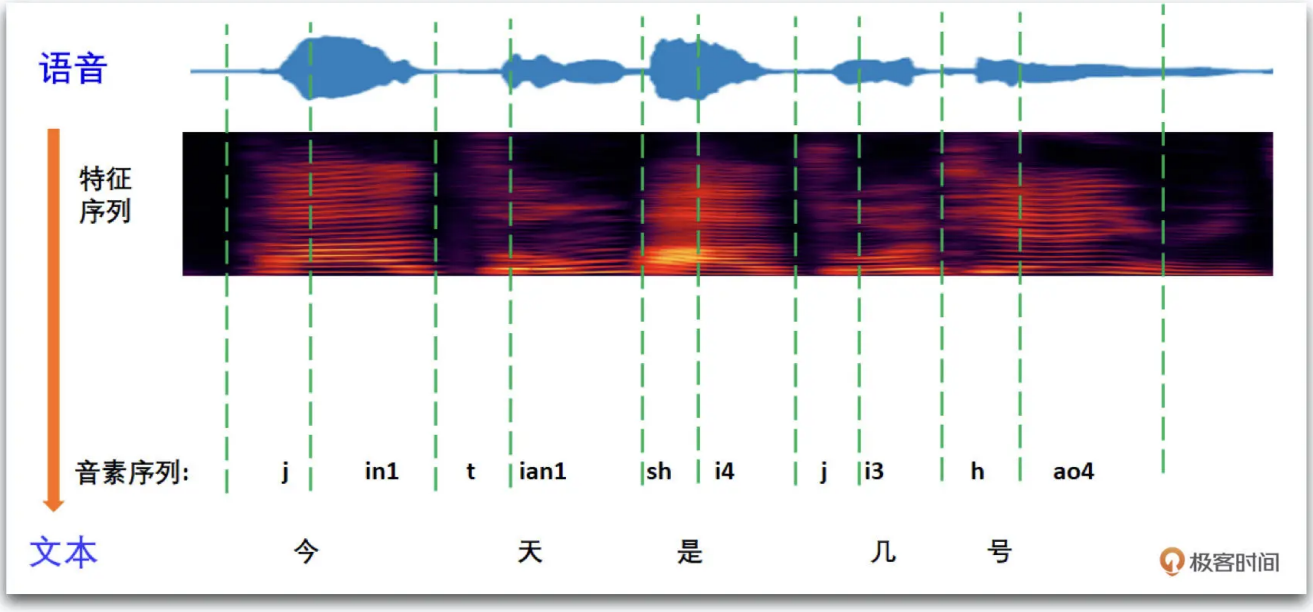


图2 ASR算法的基本原理

ASR 在音频领域的研究一直都是比较火热的方向。目前在工业界，使用的最多的是基于 Kaldi 开源框架的算法，有兴趣的话可以通过链接了解一下。Kaldi 模型以及其改进版本有很多，这里我主要介绍一下常见的 ASR 模型的构建方法。

如图 3 所示，为了实现比较准确的 ASR 系统，我们需要构建两个主要的模型：**声学模型（Acoustic Model）和语言模型（Language model）**。然后通过语言解码器和搜索算法（Ligusitic Decoding and search algorithm），结合声学模型和语言模型的结果，综合选择出概率最大的文字序列作为识别的输出。其中声学模型主要是通过语音数据库训练得到，而语言模型则主要是通过文本数据库训练得到。

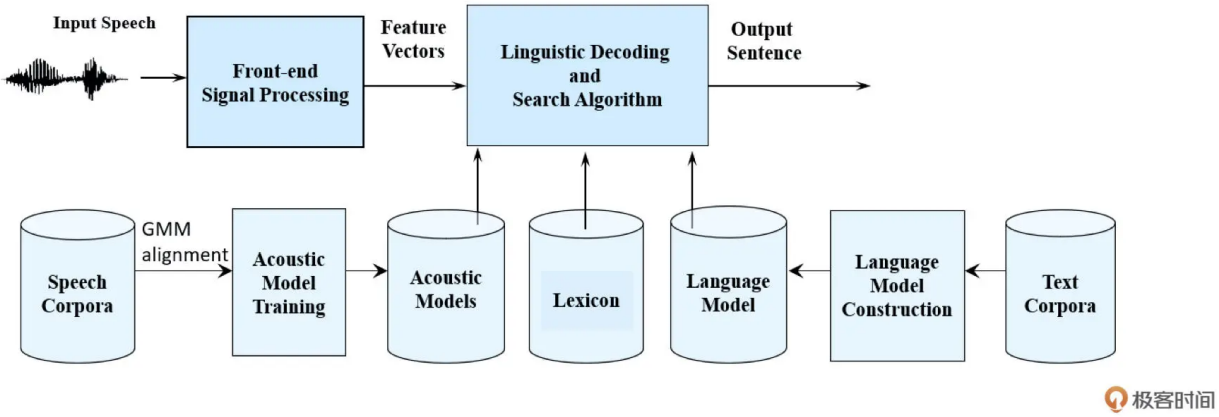


图3 ASR模型构建方法

这里你可能会有疑问，为什么我们不能从音频的特征信息直接得到文字输出，而需要这么一套相对复杂的模型系统呢？

这是因为同音字、同音词、谐音、连读等发音特性，可能导致很多容易混淆的结果，从而同一段语音可能会得到多个备选的文字方案。比如图 2 中根据音素序列可以得到“今天是几号”，也可能是“今天十几号”或者“晴天是几号”等。这时除了声学模型音素读取需要较高的准确性外，还需要语**言模型根据上下文的语境来对 ASR 的结果进行修正**。

其实，最近这几年端到端的 ASR 的研究也有很多不错的进展。比如ESPNet 开源项目里就整合了许多基于 CTC、Transformer 等技术的端到端开源模型，有兴趣的同学可以通过链接自行了解一下。

TTS

好的，了解了 ASR 的基本原理后，我们再来看看文字转语音（TTS）是如何实现的。

其实语音合成作为 ASR 的逆过程，实现起来主要是先通过一个模型把文字转为语音的特征向量，比如 MFCC，或者基频、频谱包络、能量等特征组合的形式，然后再使用声码器（Vocoder）把语音特征转换为音频信号。

那说到 TTS，就不得不提及 Google 发表的两篇重要论文，一个是WaveNet 声码器，另一个是声纹识别到多重声线语音合成的迁移学习。其中 WaveNet 声码器首次把语音合成的音频结果提升到了和真人说话一样的自然度，而 VPR 结合 WavNet 则是实现了端到端的文字到语音的生成。你可以在链接中找到这两篇论文。

WaveNet 的基本思想是利用因果空洞卷积这种自回归的 AI 模型，来逐点实现音频时域 Wave 信号的生成。用通俗一点的话来说就是，**WaveNet 是逐个采样点生成音频信号的**，也就是说生成出的第 k 个点是第 k+1 个点的输入。这种算法极其消耗算力，比如生成一秒 48kHz 采样的音频，需要循环调用 AI 模型计算 48000 次才能完成，但这样做得到的音频的效果自然度还是很不错的。

为了缩减算力，同时保持模型生成的计算速度，人们后续对其做了一系列的改进。于是就有了后来的 Fast WaveNet、Parallel WaveNet、WaveRNN、WaveGlow 和 LPCNet 等。

如图 4 所示，Fraunhofer IIS 和 International Audio Laboratories Erlangen 就曾联合发布过一篇回顾的论文，主要分析了不同的 Vocoder 的 MUSHRA 评分。

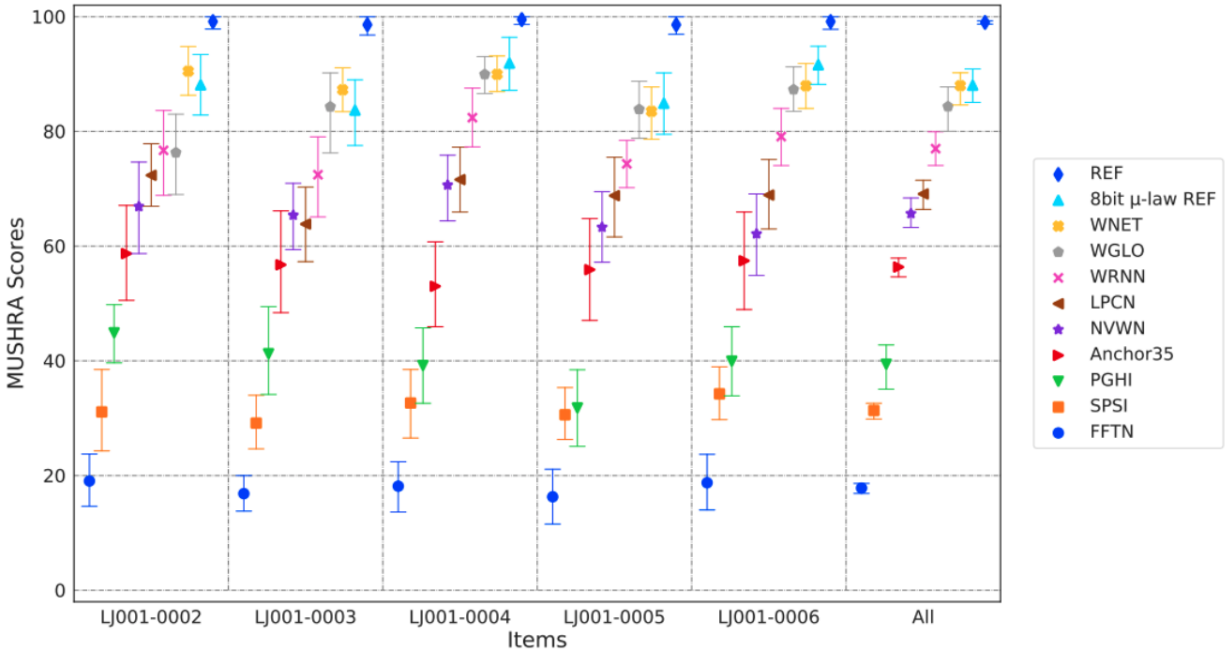


图4 不同Vocoder的MUSHRA评分

在实际使用中，为了追求效果，可以使用 WaveRNN 或者 WaveGLOW 这样效果比较好的模型。但这些模型在服务器端部署还行，在移动端部署算力还是过大。移动端上 LPCNet 或者基于传统算法的 World Vocoder 是目前比较可行的 TTS 实现方式。

好的，讲完声码器我们再看看**多重声线语音合成的基本原理**。如图 5 所示，要实现带有任意说话人音色的语音生成，需要依赖 3 个主要的模块，也就是 Speaker Encoder、Synthesizer 和 Vocoder 这三个模块。

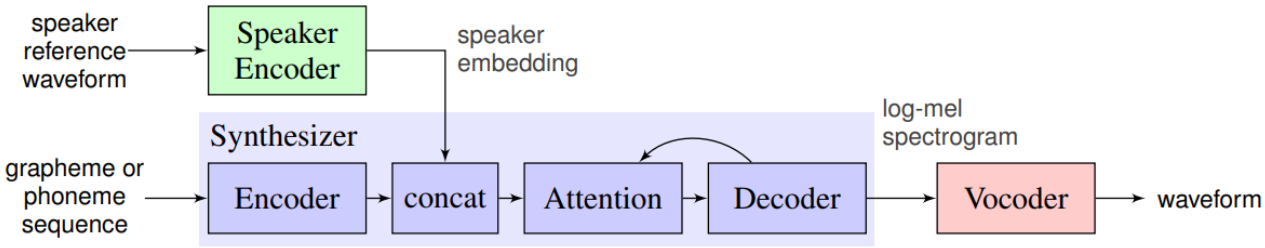


图5 声纹识别到多重声线语音合成的迁移学习
图片来源 https://arxiv.org/abs/1806.04558

其中说话人的音色和发音习惯等声纹信息，可以通过一段事先准备好的说话人语料和一个 Speaker Encoder（说话人编码器）来提取。说话人编码器其实就是我们之前说的 VPR 声

纹识别技术，常见的 VPR 技术有 I-vector、x-vector、GE2E、Deep Speaker、RawNet 等。

VPR 主要目的就是把说话人的声音特点编码成固定长度的向量（SpeakerEmbeding）。好比我们一般用指纹来验证身份，而 VPR 得到的声纹也可以用于身份识别。所以 VPR 技术其实除了可以用于 TTS 和变声，也可以用于实现身份鉴定、声纹锁等功能。

Synthesizer（合成器）则是通过音素和声纹信息合成出声码器所需的语谱特征。由于图 5 中使用的是 WaveNet 系列的 Vocoder，所以需要的是对数梅尔谱（log-mel spectrogram）作为声码器的输入。合成器也有一些现成的框架可以使用，比如 Google 的 Tacotron、微软的 Fastspeech 等。其实合成器的原理都大致相同，就是**利用 AI 模型对音素序列进行编码，然后和声纹特征融合在一起，再通过 Decoder 模型得到声码器需要的输入特征。**

VC

好的，在知道了 ASR 和 TTS 的基本原理以后，现在我们来重新审视一下我们要做的 AI 变声任务。

假如你想把自己的声音变成目标 A 的声音，只需要经过以下 6 个步骤：

1. 录制一段 A 的声音；
2. 通过 VPR 得到一个 A 的声纹（Speaker Embeding）；
3. 录制一段自己的声音；
4. 用自己的声音通过 ASR 得到音素序列；
5. 利用 TTS 的合成器把 A 的声纹和 ASR 得到的音素序列转换为声码器需要的特征；
6. 利用声码器得到变声后的音频。

实际上利用上面的技术，你只需要利用 VPR 提取任意目标的声音，就可以灵活地实现任意目标声音的转换。但要实现这么一整套 VC 系统，你需要同时实现 ASR、TTS 这两套可以说是音频领域最为复杂的 AI 模型。那和很多 AI 模型一样，你可能会想有没有什么办法可以实现端到端的变声呢？

答案是有的，尤其是**基于 GAN（Generative Adversarial Networks）技术的变声模型**，例如Cyclegan、Stargan 等模型，都在变声领域实现了端到端的变声方案，有兴趣的同学可以通过文稿中的链接了解一下。

小结

好的，今天的课程到这里就要结束了。我们来回顾一下这节课的内容。

相比于传统算法，基于 AI 模型的变声可以动态地根据发音的内容来对语音进行调整，从而实现更为自然且逼真的变声效果。基于 AI 的变声模型主要包括了三个模块：ASR、TTS 和 VPR。

Kaldi 是目前最流行的 ASR 框架。Kaldi 利用声学模型和语言模型来综合分析提取语音中的音素和文字信息，从而提升了文字识别的鲁棒性问题。而常见的端到端模型框架可以采用 ESPNet 等开源框架，在实际使用中，目前同等准确性和内容覆盖率的前提下，端到端模型和 Kaldi 框架比较起来算力会更大一些。而且端到端模型如果要针对某个场景做定制时，因为要重新训练模型，所以受到数据的限制可能会更大一些，所以端到端模型目前还没有得到大规模的使用。

TTS 技术中，WaveNet 声码器已经可以让合成语音得到媲美真人声音的自然度。而基于 VPR 声纹的 TTS 技术，可以实现语音音色根据目标声纹来任意替换，这为后续的变声应用提供了一种比较方便的实现方式。

最后，变声技术把 ASR、TTS 和 VPR 技术融合起来，从而实现了高质量的变声系统。在实践中，目前这种基于 ASR、TTS 和 VPR 的变声系统，效果要优于传统音效算法和端到端的 AI 变声算法。但这套系统的算力、存储空间的复杂性还是很高的，在部署和成本方面仍然有很多挑战。

思考题

这里留给你一道思考题，其实变声技术要实现实时的变声，依旧是一件极具挑战的事情，主要是需要在延迟和效果中做平衡。你可以思考一下，为什么离线的变声效果会优于实时在线变声，或者换句话说，为什么不引入延迟变声的效果就会变差呢？

你可以把你的答案和疑惑写下来，分享到留言区，与我一起讨论。我们下节课再见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 1 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 加餐 | 音频技术漫谈之好声音是怎么炼成的？

下一篇 结束语 | 选择比努力更重要

精选留言 (1)

写留言

晓龙
2021-12-27

为什么不引入延迟变声的效果就会变差呢？

因为说话有语境，有上下文，说话不止需要前参考，还需要后参考，人在说话时，大脑已经预先处理了，说出来就很自然，但是机器不知道你的下一句是什么？所以只能依据前参考，预测后参考，有偏差，自然度就比较差。

展开